CS100M Spring 2006 Project 4 due Thursday 3/30 at 6pm

Turn off the file backup feature in DrJava—it causes problems on some system configurations! Go to menu item **Edit→Preferences**, choose the last category, "Miscellaneous," then *uncheck* the box for "Keep Emacs-style Backup Files."

Submit your two files `Calculator.java` and `SimpleComplex.java` on-line in CMS under Project 4 before the project deadline. For java code be careful to submit the **.java** file, not the **.class** file. Both correctness and good programming style contribute to your project score.

You must work either on your own or with one partner. You may discuss background issues and general solution strategies with others, but the project you submit must be the work of just you (and your partner). If you work with a partner, you and your partner must register as a group in CMS and submit your work as a group.

Objectives

In this project, you will learn to write Java programs in the "procedural" style—the way we have written MATLAB programs up to this point. (So this is not object-oriented programming yet!) You will start with a simple program with one class and one method only to work with arithmetic operators, methods in the `Math` class, type conversion, and printing. The second question deals with *complex numbers*; and you will work with the selection statement, loops, and writing and calling `static` methods.

1 Calculator

Write a program `Calculator.java` (the class name is Calculator) to perform the following operations and print the results. The output should be "labeled." For example, the output format for part (a) below may be `a: 294`

a. $(75 - 33) \times 7$ (here $\times$ denotes multiplication).

b. 526/23

c. 526/23.0

d. The remainder of 1.15/0.25

e. The whole number (integer) portion of 39/10

f. The remainder of $-7/10$

g. Determine the whole number (integer) portion of the quotient 15/3.33. Hint: Use a cast.

h. Evaluate is "56 equal to 11." (The corresponding code is 56 == 11)

i. Evaluate the expression "56 is not equal to 11"

j. Evaluate the expression "$(\sqrt{3.5})^2$ is equal to 3.5" (Use method `Math.sqrt`)

k. The bigger value between 10^{15} and 15^{10} (Use methods `Math.pow` and `Math.max`)

l. $e^{\pi i} + 1$ (Note: $e^{x+yi} = e^x(\cos(y) + \sin(y)i)$). You must use `Math.cos` and `Math.PI`. Hint: you don't need `Math.sin` or the complex value i!

m. Generate a random number in the range of $(-e^{-7}, \log_{10}(10^3 + 1)]$. Use method `Math.random`

n. Randomly select one number from the set $\{3, 20, 1976, 30\}$ with equal probability for each value. Use method `Math.random` and a selection structure.

Compute and print the answers to all the above questions in the same program. Submit file `Calculator.java` in CMS.

2 Simple Complex Number Calculator

In this program you will learn a little about the complex number system and how to compute within it. A complex number is usually written as $3 + 4i$, or $\cos(\pi) + \sin(\pi)i$. Note that i is merely a symbol and one can think of its juxtaposition by a number as multiplication. You may know that one may assign the meaning that i is $\sqrt{-1}$. However, in this project we will not be concerned with i's value but instead we will think of a complex number as an ordered pair. So the above examples would be $(3, 4)$ and $(\cos(\pi), \sin(\pi))$, respectively. Note that the order is important: (3,4) is not the same as $(4, 3)$. The first number of the pair is called the real part of the complex number, while the second is called the imaginary part.

Your job is to build a crude complex number calculator. The code must loop repeatedly, performing the user's complex bidding: complex addition, complex division, find a complex number's magnitude or find a complex number's angle. Below is an example run of the calculator:

```
               Simple Complex Calculator

                          -i-

                         "add"
                         "div"
                         "mag"
                         "ang"
              Enter anything else to quit.


Operation?  :  Operation is add
First complex number x+yi?  x = ? x = -1.0, y = ? y = 3.1415

Second complex number u+vi?  u = ? u = 1.0, v = ? v = -3.1415
( -1.0 + 3.1415 i ) + ( 1.0 - 3.1415 i ) = ( 0.0 + 0.0 i )

Operation?  :  Operation is div
First complex number x+yi?  x = ? x = -1.0, y = ? y = 0.0

Second complex number u+vi?  u = ? u = 0.0, v = ? v = 1.0
( -1.0 + 0.0 i ) / ( 0.0 + 1.0 i ) = ( 0.0 + 1.0 i )

Operation?  :  Operation is div
First complex number x+yi?  x = ? x = 1.0, y = ? y = 0.0

Second complex number u+vi?  u = ? u = 0.0, v = ? v = -1.0
( 1.0 + 0.0 i ) / ( 0.0 - 1.0 i ) = ( 0.0 + 1.0 i )

Operation?  :  Operation is div
First complex number x+yi?  x = ? x = 0.0, y = ? y = 1.0

Second complex number u+vi?  u = ? u = -1.0, v = ? v = 0.0
( 0.0 + 1.0 i ) / ( -1.0 + 0.0 i ) = ( 0.0 - 1.0 i )

Operation?  :  Operation is mag
First complex number x+yi?  x = ? x = 3.0, y = ? y = -4.0
|( 3.0 - 4.0 i )| = 5.0

Operation?  :  Operation is ang
First complex number x+yi?  x = ? x = -1.0, y = ? y = -1.0
ang( ( -1.0 - 1.0 i ) ) = 0.7853981633974483

Operation?  :  You chose to stop or gave a bad operation.  Goodbye.
```

The file `SimpleComplex` is a skeleton of the program and contains several methods. The `main` method is the driver of the program and some parts need to be completed by you: printing the menu of operations; getting user input; performing the operations (by calling the appropriate methods). We have provided three other completed methods for you to use and to serve as examples: `add`, `isBinaryOp`, `isBadOperation`. Read these provided methods! You need to implement (write code according to specifications) four methods: `print` for displaying a complex number; `div` for division; `mag` for finding the magnitude; and `ang` for finding the angle.

Read the skeleton, compile it, and run it before you start! When you run the given incomplete program, you will see a prompt for the operation. You can type `add` to perform the addition operation but of course you won't see a correct result because you have yet to complete the code. Type anything other than the four allowed operations to quit the program. As we have provided you with code that compiles and runs, *we expect that your submitted program will at least compile*, even if there may be some logical errors. A program that does not compile will receive a significant penalty. This means you should *test* your program *as you complete the individual sections and methods*! Do not wait until you have written everything before you start testing.

Specifications

- Do not change the provided code except as marked. Write your code between the comment lines

  ```
  //----- Write your code below -----

  //----- Write your code above -----
  ```

  in method `main` and in the methods where the method body is marked by the comment

  ```
  //---- Implement this method ----
  ```

  You may use only methods in the `SimpleComplex` class and in the standard libraray (e.g., `Math` methods, method `println` from the `System` class, etc.).

- The code must loop repeatedly until the user gives a command that is not one of the expected operations. On each loop iteration your code must prompt the user for input: operation type and complex number(s). Lastly the code will display what has been computed.

- The calculator must perform the following functions for complex numbers $z = (x, y)$ and $w = (u, v)$:
  - Addition: $z + w = (x + u, y + v)$
  - Division: $\frac{z}{w} = \left(\frac{xu+yv}{u^2+v^2}, \frac{yu-xv}{u^2+v^2}\right)$
    Check for division by zero! In that case display a message stating the problem.
  - Magnitude of z: $|z| = \sqrt{x^2 + y^2}$
  - Angle of z: $\theta(z) = \tan^{-1}\left(\frac{y}{x}\right)$

- The user will input a string to indicate which operation they want to perform. Use the string "add" to signify complex addition, "div" to signify complex division, "mag" to signify complex magnitude determination, and "ang" to signify complex angle determination.

- The code will display the result using the i-notation using regular mathematical symbols. For instance if the user inputs "div", $(3, 4)$ and $(0, 1)$, then the program computes

  $$\frac{(3, 4)}{(0, 1)} = (4, -3).$$

  So the output should look like `( 3 + 4 i) / ( 0 + 1 i ) = ( 4 - 3 i )`. Note that we need not make special rules for when 0s or 1s are used, but we do need to use parentheses to denote the order of opertations. If the imaginary part is negative, the display should not include both symbols `+ -`. Note in the example above the quotient is displayed as (4 - 3 i) and not (4 + - 3 i). Do not be concerned about the number of decimal places.